**Lab Assignment 8**

**(Week 5 – Lab C and Lab D)**

Q1. You have been given a binary tree BT (may or may not be a complete binary tree). Propose an efficient scheme to find whether BT is binary min-heap or not and implement it.

Q2. You have been given N arrays of equal length (say M). Elements of these arrays are arranged in ascending order (see the example). Propose an efficient scheme to merge these arrays to create a single sorted array of length M x N

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 5 | 10 | 12 | 15 | 20 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 3 | 17 | 19 | 23 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4 | 9 | 13 | 18 | 21 |

Single sorted array:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 3 | 4 | 5 | 9 | 10 | 12 | 13 | 15 | 17 | 18 | 19 | 20 | 21 | 23 |

Q3. You have been given a binary min-heap (stored in an array). It is desired to update the Kth indexed element of this binary min-heap (*i.e.* element at index number K of the array is updated). Propose an efficient scheme to check whether the given array is still a binary min-heap or not, if not then make it as min-heap. Also, implement your logic.

Q4. You have been given a matrix of order M x N. Elements of the given matrix are arranged as follows: elements of the even indexed rows (0, 2, 4 etc.) are arranged in ascending order in respective rows and elements of the odd indexed rows (1, 3, 5 etc.) are arranged in ascending order in respective rows. Write a program to sort (in ascending order) the entire matrix in row order major (*i.e.* last element (Nth element) of the ith row is smaller than the first element of the (i+1)th row and elements in ith row are in ascending order).

Q5. You have been given an array of randomly arranged N (N>100) elements. Write an efficient program to display (not necessarily in sorted order) the 10 smallest elements in the array.

Q6. You have been given two perfect binary min-heaps (A perfect binary min-heap is a perfect binary tree in which all interior nodes have two children and all leaves have the same depth or same level and elements of the perfect binary tree satisfies the property of min-heap). Write an efficient program to merge the given two perfect binary min-heaps and construct a single binary min-heap.

Q7. A ternary tree is tree where each node can have three branches/children and in complete ternary tree, children are inserted in following order: left child, middle child, and right child. So a ternary min-heap is a complete ternary tree which satisfies the property of min-heap (parent node data is smaller than all the three children data, if present). Further, a perfect ternary min-heap is a perfect ternary tree in which all interior nodes have three children and all leaves have the same depth or same level and elements of the perfect ternary binary tree satisfies the property of min-heap. Given two perfect ternary min heap, write an efficient program to merge the given two perfect ternary min-heaps and construct a single ternary min-heap